

Applicant : David J. Boothby et al.  
Serial No. : 09/840,403  
Filed : April 23, 2001  
Page : 16

Attorney's Docket No.: 05110-009003

REMARKS

Attached is a marked-up version of the changes being made by the current amendment.

Pursuant to 37 C.F.R. 1.96, applicants hereby submit a microfiche appendix of the computer source listings included as paper appendix in the above-identified application.

Applicants request that the paper appendix be stricken from the record.

Applicants have replaced Figs. 4-8 with Pseudocode Listings 1-5, and have amended the specification to make it conform with this. Applicants submit that no new matter has been added.

Please apply any other charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date:

7/23/01

G. Roger Lee  
Reg. No. 28,963

Fish & Richardson P.C.  
225 Franklin Street  
Boston, MA 02110-2804  
Telephone: (617) 542-5070  
Facsimile: (617) 542-8906

20279811

B

**Version with markings to show changes made**

In the specification:

Replace the paragraph beginning at page 1, line 6 with the following:

-- An appendix [(appearing now in paper format to be replaced later in microfiche format)] forms part of this application. The appendix, which includes a source code listing relating to an embodiment of the invention, includes [ ] 153 frames on [ ] 2 microfiche. --

Replace the paragraphs beginning at page 9, line 17 through page 10, line 8 with the following:

-- We will describe two embodiments of a distributed synchronization program. We will first describe in general terms the overall structure of the distributed synchronization program in reference to Figs. 2 and 3 which is common to both embodiments. We will then describe [then] the first and second embodiments performing a distributed synchronization in reference to [Figs. 4-8] Pseudocode Listings 1-5.

Fig. 2 shows the relationship between the various modules of an embodiment of a distributed synchronization program. Translation Engine 1 comprises a Control Module 2 that is responsible for controlling the synchronizing process by instructing various modules to perform specific tasks on the records of the two databases being synchronized. The Control Module 2 also provides data that affects the specific operation of the various components of the synchronization program, such as the name of the databases being synchronized and user preferences. [Fig. 4] Pseudocode Listing 1 is the pseudocode of the steps taken by this module. The Synchronizer 15 has primary responsibility for carrying out the core synchronizing functions. It is a table-driven code which is capable of synchronizing various types of databases whose characteristics are provided by control module 2. The Synchronizer creates and uses a host workspace 16 (shown in detail in Fig. 3), which is a temporary data array used during the synchronization process. --

Replace the paragraph beginning at page 11, line 1 with the following:

B

[Fig. 4] Pseudocode Listing 1 is the pseudocode for the operation of Control Module 2 of the Translation Engine 1. We will use this pseudocode to generally describe distributed synchronization according to the invention. Control Module 2 first initializes itself and specifies the current user options to various modules (Step 401). In step 402, control module 2 instructs the Synchronizer to load host history file 19. Synchronizer 15 in response creates host workspace 16 data array and loads host history file 19 into host workspace 16. Host history file 19 is a file that was saved at the end of last synchronization and contains records representative of the records of the two databases at the end of the previous synchronization. Typically, the host history file contains a copy of the results of the previous synchronization of the synchronized records of the two databases. It should be noted that the content of the records of the history file may be limited only to those fields that are synchronized and the data may be translated and stored in a format different than that of the remote database or the host database. This data can be used to reconstruct the content of the records of the remote database as they were at the end of the previous synchronization. The host history file is generally used to determine changes to the databases since a previous synchronization and also to recreate records not sent from the remote segment, as will be described in detail below. If no history file from a previous synchronization exists or the user chooses to synchronize without using the history file, in step 402 the synchronizer does not load a history file. In that case, all the records from both databases will be loaded into the host workspace. We will describe the rest of the operation of the control module as if a history file exists and will be used. --

Replace the paragraph beginning at page 12, line 7 with the following:

-- Control Module 2 then instructs remote segment to send the records of the remote database (step 404). Remote segment 26 reads the remote database records and sends them to Synchronizer 15 for writing into the host workspace. The actions taken by the synchronizer and the remote segment in response to step 404 will be described in detail in reference to [Figs. 5, 6, 7, and 8] Pseudocode Listings 2-5, below. --

Replace the paragraphs beginning at page 14, line 14 through page 15, line 16 with the following:

-- If the user chooses to proceed with synchronization and to unload, the records are then unloaded in order into the host database, the remote database and the History File. The Synchronizer in conjunction with the host translator and the remote segment perform the unloading for the databases. Synchronizer 15 creates a host history File and unloads the records into it. Control Module 2 first instructs the host translator to unload the records from host workspace into the host database. Following unloading of the host records, Control Module 2 instructs the synchronizer and the remote segment to unload the remote records from the host workspace (step 409). We will describe in detail below, in reference to [Figs. 5-8] Pseudocode Listings 2-5, the specific actions taken by Synchronizer 15 and remote segment 26 in order to unload data from the host workspace into the remote database and the update remote history file 28. Control Module 2 next instructs the Synchronizer to create a new History File (step 112). At this point Synchronization is complete.

Referring to [Figs. 5 - 8] Pseudocode Listings 2-5, we will now describe the actions taken by the remote segment in coordination with the Synchronizer in response to the instructions from control module 2 in step 404 to load records of the remote database and in step 409 to unload the records of the remote database from the host workspace. Specifically, we will describe two embodiments. In the case of the first embodiment, the remote database assigns unique identification codes (i.e. unique IDs) to each of its records as they are created. In the case of the second embodiment, the remote database does not assign unique IDs to its records. [Fig. 5] Pseudocode Listing 2 is the pseudocode for the steps taken by the remote segment while [Fig. 6] Pseudocode Listing 3 is the pseudocode for the steps taken by the Synchronizer in the case of the second embodiment. Similarly, [Fig. 7] Pseudocode Listing 4 is the pseudocode for the steps taken by the remote segment while [Fig. 8] Pseudocode Listing 5 is the pseudocode for the steps taken by the Synchronizer in the case of the first embodiment. --

Replace the paragraphs beginning at page 17, line 18 through page 18, line 8 with the following:

-- Having described in general terms the actions taken by the remote segment in coordination with the Synchronizer in response to the instructions from control module 2 in steps 404 and 409 [(Fig. 4)] Pseudocode Listing 1, we will now describe in detail a first embodiment

B

of their operation for the case where the remote database assigns unique IDs to its records. We will do so in reference to [Figs. 5 and 6] Pseudocode Listings 2-3.

[Fig. 5] Pseudocode Listing 2 is the pseudocode for steps taken by the remote segment in response to the instruction by control module in step 404 to load the remote database records into the host workspace [(Fig. 4)] Pseudocode Listing 1. The remote segment first initializes (i.e. creates) a remote workspace in the remote computer (step 501). The remote segment then compares the name of the host history file with the name of any remote history file in the remote computer. If the remote segment finds a remote history file that matches the host history file (i.e. a remote history file that matches the host history file) (step 502), then the remote segment examine the date and time stamp of the host and remote history files. If the date and time stamp in the remote history file matches the one in the host history file (step 503), then the remote segment determines that two history files correspond to one another. Hence, the remote segment loads the remote history file into the remote workspace. --

Replace the paragraph beginning at page 23, line 10 with the following:

-- After the remote segment has finished providing data to the host segment, the host segment synchronizes the two databases based on the input from the remote segment. The remote segment waits until the host segment finishes synchronizing and instructs the remote segment in step 409 in [Fig. 4] Pseudocode Listing 1 to begin unloading into the remote database (step 532). --

Replace the paragraphs beginning at page 25, line 6 through page 28, line 5 with the following:

-- Referring to [Fig. 6] Pseudocode Listing 3, in the described embodiment, all records received by the host segment from the remote segment are flagged with one of Added, Changed, or Deleted flags. For all records received from the remote segment (step 601), the host synchronizer performs the following functions. If a received record is flagged as an added record (step 602), then the received record is added to the host workspace (step 603). Since the record is new, it is not associated or linked to any history file record. If a record is flagged as a "changed" record (step 604), then the Synchronizer uses the received unique ID to find the corresponding record in the history file (step 605) and links the received remote record to that

B

history file record (step 606). If the received record is flagged as a "deleted" record (step 607), then the Synchronizer uses the received unique ID to find the corresponding record in the history file (step 608) and marks the history file record as deleted (step 609).

After all the received records are analyzed (step 611), if any host history file records containing remote database unique IDs are left that were not matched against the received records, the synchronizer assumes that those records represent the remote database records that are unchanged. For all those records (step 612), the synchronizer clones the host history file record (i.e. create a workspace entry and copy all the host history file record in to that entry) and treats it as a record received from the remote database. At this point the host segment proceeds with synchronization since the records of the remote database have now been loaded. In essence, referring back to [Fig. 4] Pseudocode Listing 1, this is the end of step 404.

As previously described, after the synchronizer has performed CAAR, the user must confirm to proceed with updating the remote database (step 406 in [Fig. 4] Pseudocode Listing 1). If the user decides to terminate the synchronization, changes are not made to the host history file or the databases. In the case of the remote database, as described in reference to [Fig. 5] Pseudocode Listing 2, the remote segment is waiting for the synchronizer to finish synchronizing. If the user aborts synchronization (step 533), the remote segment discards the remote workspace (step 534), saves the original history file without any changes (step 535), and terminates the process at the remote computer.

If the user confirms to proceed with updating the database (step 406 in [Fig. 4] Pseudocode Listing 1), control module 2 instructs the synchronizer and the remote segment to proceed with unloading the records from the workspace into the remote database. As stated, at this point, the remote segment is waiting for the synchronizer to finish synchronizing (step 532 in [Fig. 5] Pseudocode Listing 2). During the synchronization, the synchronizer has determined what "actions" with respect to which record in which database should be taken (update, delete, or add) to complete synchronization. If changes or additions are made to the host database in the case of particular record but no action need be taken with respect to that record in the remote database, the synchronizer determines that an "acknowledgement" should be sent to the remote segment. The synchronizer sends all the actions concerning the remote database together with the associated record to the remote (step 616). The synchronizer then sends the unique ID of

B

those records that require "acknowledgements" to be sent to the remote together with an appropriate flag (step 617).

Referring again to [Fig. 5] Pseudocode Listing 2, for each action item or acknowledgement received at the remote segment (step 538), the following steps are performed. If the received data indicates an "acknowledgement" or "action" with respect to a record that was added or changed since the previous synchronization, the remote segment marks the new workspace entry that was created in either step 520 or step 525 as acknowledged (step 540). The remote segment also discards or removes any other entry in the workspace that contains the unique ID of this record, which is typically the entry that was loaded from the remote history file. Therefore, as previously described, this entry as opposed to the old remote history file entry associated with this record will be written into the history file at the end of the process at the remote segment. This in essence updates the history file, as will be described below.

If the received data indicates an action item that tells the remote segment to update, change, or add a remote database record (step 543), the remote segment performs that action with respect to the remote database. The remote segment also performs the same steps as steps 540 and 541 (step 544 and 545). If a new record was added to the database (step 546), it will be assigned a new unique ID. The remote segment sends that unique ID to the host segment (step 547). The host segment includes that unique ID in the host work space in association with that record (step 618 in [Fig. 6] Pseudocode Listing 3).

After all the records have been received, the remote segment discards all unacknowledged entries from the workspace. Therefore, in the case of those added or changed records with which the user decided not to update the host database, the remote history file remains unchanged. The remote history file is then updated from the remote workspace. At this point the control module continues with step 410 in [Fig. 4] Pseudocode Listing 1, i.e. creating the history file to end the synchronization of the two databases. --

Replace the paragraph beginning at page 29, line 1 with the following:

-- We will describe the operation of this embodiment in reference to [Figs. 7 and 8] Pseudocode Listings 4-5. Steps 701 -711 are the same as steps 501-511 in [Fig. 5] Pseudocode

B

Listing 2, described above in reference to the first embodiment. These steps are generally concerned with finding the correct remote history file. --

Replace the paragraphs beginning at page 30, line 29 through page 32, line 28 with the following:

-- After the remote segment has finished providing data to the host segment, the host segment synchronizes the two databases based on the input from the remote segment. The remote segment waits until the host segment finishes synchronizing and instructs the remote segment in step 409 in [Fig. 4] Pseudocode Listing 1 to begin unloading into the remote database (step 724).

Referring to [Fig. 8] Pseudocode Listing 5, as in the case of the first embodiment, the synchronizer on the host computer uses the information to identify those records in the host history file that correspond to the unchanged remote database records. For every record received from the remote segment that is flagged as added (step 801), the synchronizer adds the record to the host workspace (step 802) and during CAAR compares the record to the history file to determine whether the record is a changed or added record. For every record received from the remote segment that is flagged as "unchanged" (step 804), in the same manner as the first embodiment, the synchronizer finds the corresponding host history file record by finding a record that has the same hash number as that sent by the remote synchronizer (step 805). The synchronizer then clones the record (step 806), as previously described, and treats as if it is a record received from the remote database. At the end of this process, when all the records of the remote database are loaded into the host workspace, the control module proceeds to step 405 in [Fig. 4] Pseudocode Listing 1 to begin CAAR. CAAR will then analyze the records in the host workspace to determine which remote records were added, which were changed, and which were deleted since the previous synchronization.

After CAAR, if the user confirms to proceed with updating the database, control module 2 instructs the synchronizer and the remote segment to proceed with unloading the records from the workspace into the remote database (step 409 in [Fig. 4] Pseudocode Listing 1). As stated, at this point, the remote segment is waiting for the synchronizer to finish synchronizing (step 724 in [Fig. 7] Pseudocode Listing 4). During performing CAAR, the synchronizer has determined

B



what actions should be taken (update, delete, or add) to each database. If changes or additions are made to the host database in the case of a particular record but no action need be taken with respect to that record in the remote database, the synchronizer determines that at least an "acknowledgement" is to be sent to the remote segment. The synchronizer sends all the actions concerning the remote database together with the associated record and remote workspace index to the remote (step 809). The synchronizer then sends the remote workspace index of those records that require acknowledgements to be sent to the remote together with an appropriate flag (step 810). Therefore, the remote workspace index is used to identify which records in the remote workspace should be "acknowledged".

Referring back to [Fig. 7] Pseudocode Listing 4, steps 725-729 are the same as steps 533-537, which were described in reference to the first embodiment. For each action item or acknowledgement received at the remote segment (step 730), the following steps are performed. If the data received indicates an "acknowledgement" or "action" with respect to a record that was sent to the host segment flagged as "added" (step 731), the remote segment marks the new workspace entry that was created in either step 718 as acknowledged (step 732). It should be noted that the remote workspace index number is used to locate the remote workspace entry. Therefore, as previously described, this entry will be written into the history file at the end of the process at the remote segment. --

Replace the paragraph beginning at page 33, line 3 with the following:

-- After all the records have been received, the remote segment discards all unacknowledged entries from the workspace, which were newly created entries which were not acknowledged. Therefore, in case of those added or changed records with the user decided not to update the host database with, the remote history file remains unchanged. The remote history file is then updated from the workspace. At this point the control module continues with step 410 in [Fig. 4] Pseudocode Listing 1, i.e. creating the history file to end the synchronization of the two databases. --

B